

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ  
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА  
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ  
ІНФОРМАТИКИ**

**Допускається до захисту**  
Завідувач кафедри \_\_\_\_\_ О.О. Ємець  
(підпис)  
«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО БАКАЛАВРСЬКОЇ РОБОТИ  
на тему  
ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРУ З ТЕМИ  
«ЦИКЛ WHILE В EXCEL VBA»  
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ  
«ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ» ЦИКЛУ**

**зі спеціальності 122 «Комп'ютерні науки»**

**Виконавець роботи** Срібний Сергій Юрійович

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**Науковий керівник** доц., каф.-мат. наук, Чілікіна Т.В.

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**ПОЛТАВА 2021р.**

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_ **О.О. Ємець**  
« 8 » вересня 2020р.

**Завдання та календарний графік  
виконання дипломної роботи**

**Студент(ка) спеціальності 122 «Комп'ютерні науки»  
Прізвище, ім'я, по батькові Срібний Сергій Юрійович**

1. Тема **«Програмна реалізація тренажеру з теми «Цикл while в Excel VBA»**  
затверджена наказом ректора № 121-Н від « 1 » вересня 2020 р.  
Термін подання студентом бакалаврської роботи « 20 » травня 2021 р.
2. Вихідні дані до дипломної роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.
3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

***ВСТУП***

***1 Постановка задачі***

***1.1 Постановка задачі розробки тренажера***

***2 Інформаційний огляд***

***2.1. Огляд робіт, де розглянуте аналогічне до теми роботи завдання.***

***2.2. Позитивні аспекти оглянутих робіт.***

***2.3. Вади розробок з оглянутих робіт.***

***2.4. Актуальність розробки програмного забезпечення***

***3 Теоретична частина***

***3.1 Загальні відомості***

***3.1.1 Цикл Do .. While***

***3.1.2 Цикл Do .. Until***

***3.1.3 Цикл Do .. Loop While***

***3.1.4 Цикл Do .. Loop Until***

***3.2 Алгоритм роботи тренажера***

***3.3 Блок-схема програми-тренажера***

***4 Практична частина***

***4.1 Опис програмного забезпечення***

***4.2 Дослідження можливостей програмної реалізації***

***ВИСНОВКИ***

***СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ***

***ДОДАТОК А***

4. Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

### 5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	Чілікіна Т.В.	8.09.20	8.09.20
2. Інформаційний огляд	Чілікіна Т.В.	8.09.20	8.09.20
3. Теоретична частина	Чілікіна Т.В.	8.09.20	8.09.20
4. Практична реалізація	Чілікіна Т.В.	8.09.20	8.09.20

### 6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	10.05.21	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.20	
3. Постановка задачі	1.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	2.11.20	
5. Теоретична частина	1.02.21	
6. Практична частина	17.05.21	
7. Закінчення оформлення	21.05.21	
8. Доповідь студента на кафедрі	28.05.21	
9. Доробка (за необхідністю), рецензування	14.06.21	

Дата видачі завдання « 8 » вересня 2020 р.

Студент Срібний Сергій Юрійович

Науковий керівник \_\_\_\_\_ доц., каф.-мат. наук, Чілікіна Т.В.  
(підпис)

### **Результати захисту бакалаврської роботи**

Дипломна робота оцінена на \_\_\_\_\_  
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № \_\_\_\_\_ від « \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

Секретар ЕК \_\_\_\_\_  
(підпис) (ініціали та прізвище)

## РЕФЕРАТ

**Записка:** 45 стор., в т.ч. основна частина 40 стор., джерел - 11.

**Предмет розробки** – тренажер за темою «Цикл while в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

**Мета роботи** – розробка тренажеру за темою «Цикл while в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

**Методи, які були використані для розв’язування задачі** – Робоча програма розроблена в середовищі Visual Studio 2017 на платформі UNITY 2018 на мові програмування C#.

**Ключові слова:** тренажер, цикли мови VBA, дистанційне навчання.

# ЗМІСТ

ВСТУП.....	4
1 Постановка задачі.....	6
1.1 Постановка задачі розробки тренажера.....	6
2 Інформаційний огляд .....	7
2.1. Огляд робіт, де розглянуте аналогічне до теми роботи завдання.....	7
2.2. Позитивні аспекти оглянутих робіт. ....	7
2.3. Вади розробок з оглянутих робіт. ....	7
2.4. Актуальність розробки програмного забезпечення.....	8
3 Теоретична частина.....	9
3.1 Загальні відомості .....	9
3.1.1 Цикл Do .. While .....	9
3.1.2 Цикл Do .. Until.....	10
3.1.3 Цикл Do .. Loop While.....	10
3.1.4 Цикл Do .. Loop Until .....	11
3.2 Алгоритм роботи тренажера.....	13
3.3 Блок-схема програми-тренажера.....	21
4 Практична частина .....	23
4.1 Опис програмного забезпечення .....	23
4.2 Дослідження можливостей програмної реалізації.....	29
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	41
ДОДАТОК А.....	43

## ВСТУП

Покращення мобільності під час навчання означає покращення самого навчання. Використовуючи програми-тренажери на своєму телефоні або ноутбучі в дорозі чи у будь-який інший вільний час дозволяє повноцінно навчатися чи покращувати свої знання.

Саме тому розробка та розвиток програм-тренажерів має велике значення для покращення освітнього процесу в цілому.

Мета роботи – створити тренажер для навчання студентів темі «Цикл while в Excel VBA» з дисципліни «Організація та обробка електронної інформації».

Об'єкт роботи – створення програмного забезпечення у вигляді тренажеру для систем дистанційного навчання.

Предмет роботи – програмне забезпечення для навчання темі «Цикл while в Excel VBA».

Методи роботи – Для створення програми було використано мову програмування C#, платформу UNITY 2018 та Visual Studio 2017.

Структура пояснювальної записки до бакалаврської роботи:

- титульний аркуш;
- завдання до бакалаврської роботи;
- реферат;
- зміст;
- перелік умовних позначень, символів, одиниць, скорочень і термінів;
- вступ;
- основна частина;
- висновки;
- рекомендації;
- список використаних джерел;

Обсяг пояснювальної записки: 45 стор., в т.ч. основна частина 40 стор., джерел  
- 11.

## **1 Постановка задачі**

### **1.1 Постановка задачі розробки тренажера**

Основною задачею даної бакалаврської роботи являється програмна реалізація елементів тренажеру з теми «Цикл while в Excel VBA» для дисципліни «Організація та обробка електронної інформації».

Завданням даної бакалаврської роботи є:

1. Вибір способу програмної реалізації тренажера;
2. Складання алгоритму роботи програмного забезпечення;
3. Складання блок-схеми алгоритму роботи програмного забезпечення;
4. Програмна реалізація тренажеру;
5. Перевірка валідності та тестування програмного забезпечення;

Основними вимогами до програмного забезпечення є:

1. Зрозумілість видачі інформації;
2. Покрокове пояснення;
3. Можливість виходу з застосунку через кнопку;
4. Можливість переходів між теоретичними завданнями;
5. Вивід повідомлення про помилку;



## **2 Інформаційний огляд**

### **2.1. Огляд робіт, де розглянуте аналогічне до теми роботи завдання.**

Під час дослідження робіт зі схожим завданням було знайдено:

1. Кулинич М.К. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення тренажеру з теми «Злиття впорядкованих послідовностей» дистанційного навчального курсу «Алгоритми та структури даних» [1];
2. Сивокінь О.Ю. Пояснювальна записка до дипломної роботи на тему «Логіка висловлювань» дистанційного навчального курсу «Математична логіка» та розробка його програмного забезпечення [2].

### **2.2. Позитивні аспекти оглянутих робіт.**

У роботі Кулинич М.К. було реалізовано:

1. Візуалізацію роботи над сортуванням злиттям;
2. Перевірку введеної відповіді;
3. У повідомленні з помилковою відповіддю додано підказку з правильною відповіддю.

У роботі Сивокінь О.Ю. було реалізовано:

1. Тренажер створено для андроїда;
2. Тренажер має зручний дизайн;
3. Перевірка введеної відповіді;
4. У повідомленні з помилковою відповіддю додано підказку з правильною відповіддю;
5. Існує можливість повторити роботу з тренажером не перезавантажуючи його.

### **2.3. Вади розробок з оглянутих робіт.**

У роботі Кулинич М.К. було реалізовано:

1. Відсутні теоретичні завдання;
2. Неможливо повторити роботу без перезапуску застосунку.

У роботі Сивокінь О.Ю. було реалізовано:

1. Тренажер існує лише на андроїд 7.0 і вище;

#### **2.4. Актуальність розробки програмного забезпечення**

Розглянувши схожі за темою тренажери було вирішено створити тренажер, що буде містити в собі всі можливі позитивні аспекти цих робіт та буде уникати їх вад.

Створення програмного забезпечення для навчання темі Цикл while в Excel VBA» з дисципліни «Організація та обробка електронної інформації» є актуальним та необхідним.

## 3 Теоретична частина

### 3.1 Загальні відомості

#### 3.1.1 Цикл Do .. While

Конструкція циклу, що тестує свою умову детермінанта до виконання циклу.

синтаксис:

Do While Condition

Statements

Loop

Condition - логічне вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу для перевірки умови

VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само True.

При виконанні циклу Do While спочатку тестується логічне вираз (Condition); якщо воно дорівнює True - виконується тіло циклу. При досягненні ключового слова Loop управління знову передається в початок циклу і знову перевіряється логічне вираз. Так відбувається до тих пір, поки логічне вираження не стане False. Коли логічне вираз стає False - управління передається оператору, наступному за ключовим словом Loop.

Зверніть увагу! Якщо логічний вираз дорівнює False при першому виконанні циклу Do While, то управління відразу передається оператору, наступному за Loop, а оператори, що знаходяться в тілі циклу відповідно пропускаються. Іншими словами кажучи, цикл Do While дозволяє ні разу не виконувати оператори всередині нього.

### 3.1.2 Цикл Do .. Until

Ще один цикл, який тестує умову детермінанта до виконання циклу.

синтаксис:

Do Until Condition

Statements

Loop

Condition - логічне вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу для перевірки умови

VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само False.

В іншому цикл Do Until повністю аналогічний циклу Do While.

Для тестування умов після виконання тіла циклу треба помістити логічне вираз в кінець блоку операторів, що складають тіло циклу, після ключового слова Loop, яке повідомляє про кінець циклу.

### 3.1.3 Цикл Do .. Loop While

Конструкція циклу, що тестує свою умову детермінанта після виконання циклу.

синтаксис:

Do

Statements

Loop While Condition

Condition - логічне вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само True.

При виконанні циклу Do Loop While спочатку виконуються оператори тіла циклу, потім після досягнення ключового слова Loop тестується логічне вираз (Condition); якщо воно дорівнює True - управління передається в початок тіла циклу і цикл повторюється знову. Так відбувається до тих пір, поки логічне вираження не стане False. Коли логічне вираз стає False - управління передається оператору, наступному за рядком Loop While ...

Зверніть увагу! Навіть якщо при першому виконанні циклу Do Loop While логічний вираз дорівнює False тіло циклу все одно буде виконано. Іншими словами кажучи, незалежно від значення логічного виразу, представленого за допомогою Condition, цей цикл завжди виконується, принаймні, один раз.

### **3.1.4 Цикл Do .. Loop Until**

Ще один цикл, який тестує умову детермінанта після виконання циклу.  
синтаксис:

Do

Statements

Loop Until Condition

Condition - логічне вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само False.

В іншому цикл Do Loop Until повністю аналогічний циклу Do Loop While.

Слід сказати, що представлені вище приклади дані чисто в пізнавальних цілях, щоб можна було зрозуміти саму суть організації невизначених циклів з перевіркою умов до і після виконання тіла циклу. Сама ж гнучкість невизначених циклів тут явно не проглядається.

### 3.2 Алгоритм роботи тренажера

Алгоритм роботи з тренажером описано наступними кроками:

Крок 1: Після запуску тренажера користувач переходить до головного меню тренажера.

Крок 2: Після натиснення на кнопку «Перейти до роботи» користувач отримує довідковий матеріал з теми.

Крок 3: Отримання довідкового матеріалу:

«Цикл Do .. While

Конструкція циклу, тестуються свою умову детермінанта до виконання циклу.

синтаксис:

Do While Condition

Statements

Loop

Condition - логічний вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу для перевірки умови»

Крок 4: Отримання довідкового матеріалу:

«VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само True.

При виконанні циклу Do While спочатку тестується логічне вираз (Condition); якщо воно дорівнює True - виконується тіло циклу. При досягненні ключового слова Loop управління знову передається в початок циклу і знову перевіряється логічне вираз. Так відбувається до тих пір, поки логічне

вираження не стане False. Коли логічне вираз стає False - управління передається оператору, наступному за ключовим словом Loop.»

Крок 5: Отримання довідкового матеріалу:

«Зверніть увагу! Якщо логічний вираз дорівнює False при першому виконанні циклу Do While, то управління відразу передається оператору, наступному за Loop, а оператори, що знаходяться в тілі циклу відповідно пропускаються. Іншими словами кажучи, цикл Do While дозволяє ні разу не виконувати оператори всередині нього.

Цикл While ... Wend

Цикл While ... Wend в VBA Excel призначений для повторення блоку операторів до тих пір, поки виконується задана умова (повертається значення True). За своїм функціоналом його можна вважати обмеженою версією циклу Do While ... Loop, який дозволяє перевіряти умову як до, так і після виконання операторів, і примусово завершувати цикл за допомогою оператора Exit Do. У циклу While ... Wend таких можливостей немає: умова може бути поставлено лише перед блоком операторів, а примусовий вихід із циклу не передбачений.»

Крок 6: Отримання довідкового матеріалу:

«Цикл Do .. Until

Ще один цикл, який тестує умова детермінанта до виконання циклу.

синтаксис:

Do Until Condition

Statements

Loop

Condition - логічний вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу для перевірки умови



VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само False.

В іншому цикл Do Until повністю аналогічний циклу Do While.»

Крок 7: Отримання довідкового матеріалу:

«Цикл Do .. Loop While

Конструкція циклу, тестується свою умову детермінанта після виконання циклу.

синтаксис:

Do

Statements

Loop While Condition

Condition - логічний вираз для детермінанта циклу

Statements - один, жодного або кілька операторів, які складають тіло циклу

Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

VBA виконує цикл поки логічне вираження, представлене за допомогою Condition, так само True.»

Крок 8: Отримання довідкового матеріалу:

« При виконанні циклу Do Loop While спочатку виконуються оператори тіла циклу, потім після досягнення ключового слова Loop тестується логічне вираз (Condition); якщо воно дорівнює True - управління передається в початок тіла циклу і цикл повторюється знову. Так відбувається до тих пір, поки логічне вираження не стане False. Коли логічне вираз стає False - управління передається оператору, наступному за рядком Loop While ...

Зверніть увагу! Навіть якщо при першому виконанні циклу Do Loop While логічний вираз дорівнює False тіло циклу все одно буде виконано. Іншими словами кажучи, незалежно від значення логічного виразу,

представленого за допомогою Condition, цей цикл завжди виконується, принаймні, один раз.»

Крок 9: Отримання довідкового матеріалу:

«Короткий посібник з VBA While Loops

Формат циклу	Опис	Приклад
Do While ... Loop	Запускається 0 або більше разів, поки умова виконується	Do While result = «Верно» Loop
Do ... Loop While	Запускається 1 або більше разів, поки умова виконується	Do Loop While result = «Верно»
Do Until ... Loop	Запускається 0 або більше разів, поки умова не буде виконана	Do Until result <> «Верно» Loop
Do ... Until Loop	Запускається 1 або більше разів, поки умова не буде виконана	Do Loop Until result <> «Верно»
While ... Wend R	Запускається 0 або більше разів, поки умова істинна. Примітка: цей цикл вважається застарілим.	While result = «Верно» Wend
Exit the Do Loop	завершує Do	Do While i < 10    i = GetTotal    If i < 0 Then Exit Do    End If Loop

Таблиця 3.1 – Короткий посібник з VBA While Loops

Крок 10: Після закінчення роботи з довідковими матеріалами користувач переходить до практичних завдань.

Крок 11: Отримання практичного завдання:

«Для чого використовується оператор Loop під час роботи з циклом?

Варіанти відповіді:

1. Loop - логічний вираз для детермінанта циклу
2. Loop - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

3. Loop - один, жодного або кілька операторів, які складають тіло циклу

Правильна відповідь - № 2»

Крок 12: Отримання практичного завдання:

«Для чого використовується оператор Condition під час роботи з циклом?

Варіанти відповіді:

1. Condition - логічний вираз для детермінанта циклу
2. Condition - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

3. Condition - один, жодного або кілька операторів, які складають тіло циклу

Правильна відповідь - № 1»

Крок 13: Отримання практичного завдання:

«Для чого використовується оператор Statements під час роботи з циклом?

Варіанти відповіді:

1. Statements - логічний вираз для детермінанта циклу
2. Statements - один, жодного або кілька операторів, які складають тіло циклу

3. Statements - ключове слово, вказує на закінчення тіла циклу і позначає місце, з якого VBA повертається в початок циклу після перевірки умови

Правильна відповідь - № 2

Крок 14: Отримання практичного завдання:

«Виконання циклу продовжується поки значення Condition =?

Варіанти відповіді:

1. Error
2. False
3. True

Правильна відповідь - № 3»

Крок 15: Отримання практичного завдання:

«Вкажіть правильний синтаксис циклу While Wend

Варіанти відповіді:

1. Wend condition

[ statements ]

While

2. While condition

[ statements ]

Wend

3. While statements

[condition]

Wend

Правильна відповідь - № 2»

Крок 16: Отримання практичного завдання:

«Вкажіть синтаксис циклу Do While... Loop з перевіркою умови до виконання операторів?

Варіанти відповіді:

1. Do While condition

[ statements ]

[ Exit Do ]

[ statements ]

Loop

2. Do While condition

[ statements ]

Loop

3. Do While condition

[ statements ]

[ Exit Do ]

Loop

Правильна відповідь - № 1»

Крок 17: Отримання практичного завдання з полем для вводу:

«Вкажіть синтаксис циклу Do While... Loop з перевіркою умови до виконання операторів?

Результатом є наступний код:

«Do

[ statements ]

[ Exit Do ]

[ statements ]

Loop While condition»

Крок 18: Отримання практичного завдання з полем для вводу:

«Запишіть синтаксис для команди Exit Do

Результатом є наступний код:

«If n = 1000 Then

Exit Do»

Крок 19: Після закінчення роботи з тренажером користувач може повторити роботу з тренажером після натиснення на кнопку «Повтор?».

### 3.3 Блок-схема програми-тренажера

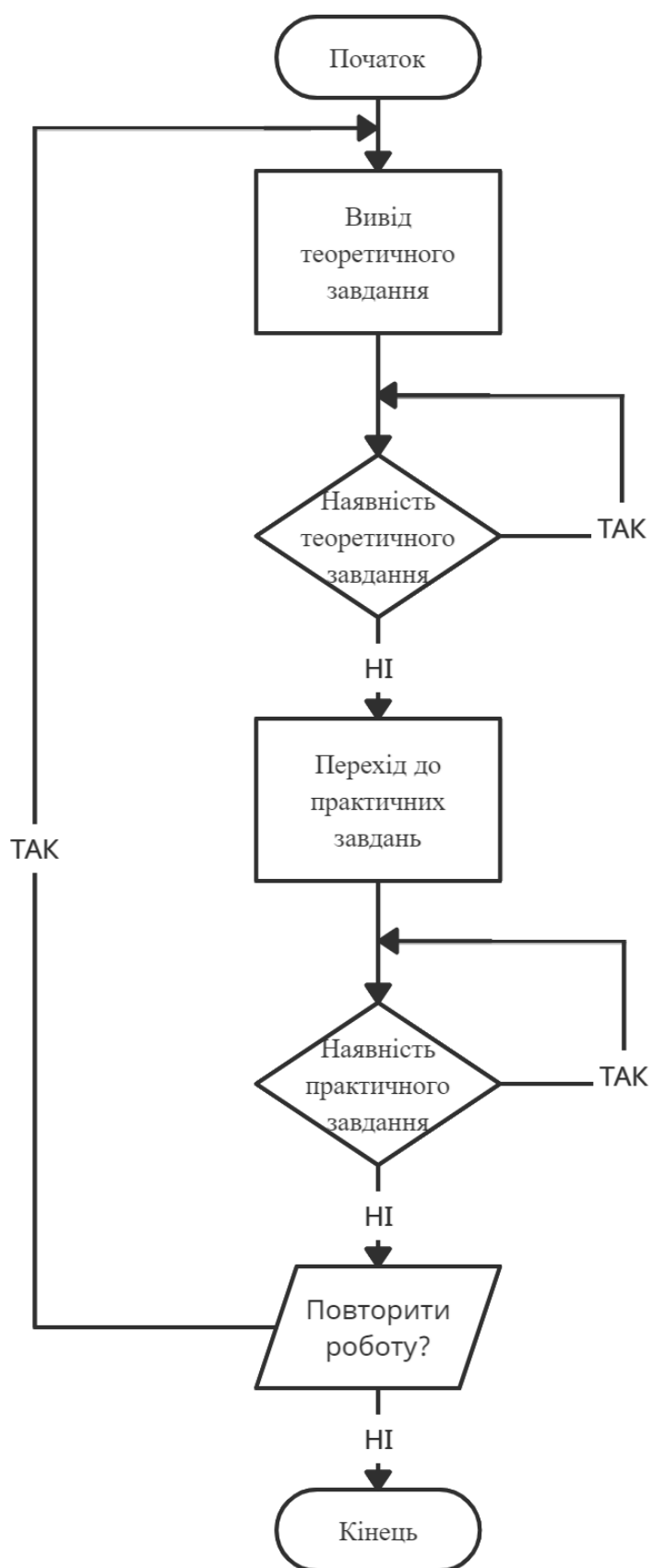


Рисунок 3.1 – Блок-схема програми-тренажера

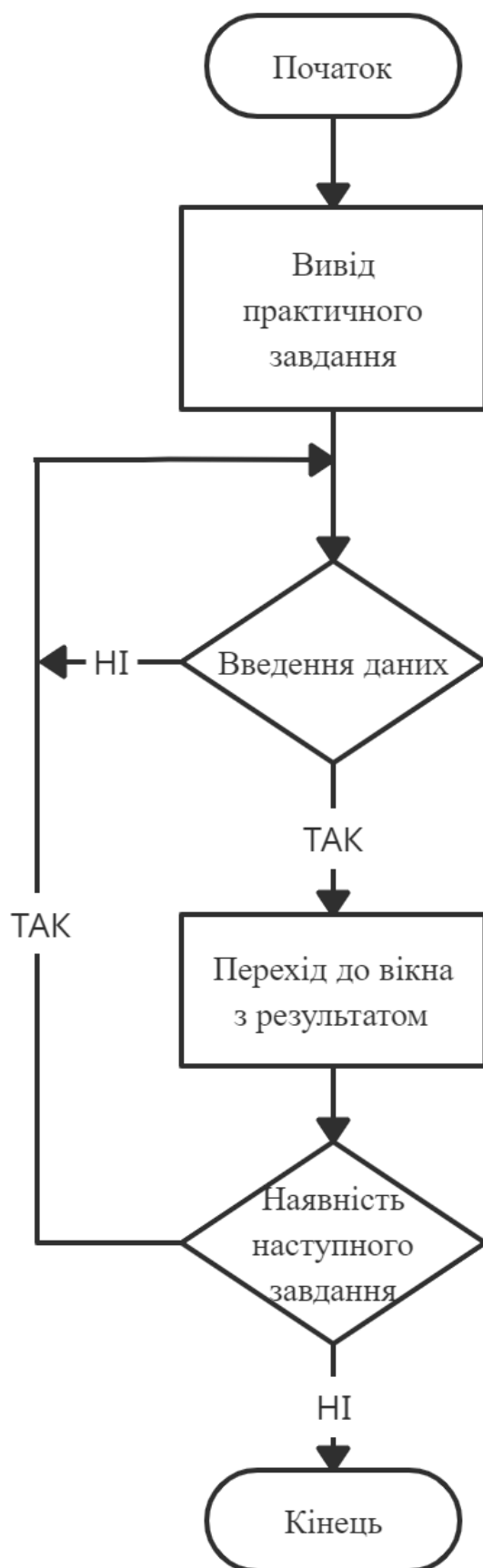


Рисунок 3.2 – Блок-схема роботи з практичним матеріалом



## 4 Практична частина

### 4.1 Опис програмного забезпечення

Програма-тренажер була розроблена за допомогою середовища розробки Visual Studio 2017 та UNITY 2018. Робота з кнопками передбачена в платформі UNITY 2018. Їх роботу можна описати як закриття відображення одного об'єкту та відкриття наступного. Реалізовано кнопки переходу між завданнями теоретичного та практичного типу та повернення до головного меню.

Код скрипту для переходу між об'єктами в UNITY 2018:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input; //об'явлення об'єктів
    public GameObject AfterInput; //об'явлення об'єктів

    public void Next()
    {
        Input.SetActive(false); //вимкнення першого об'єкту
        AfterInput.SetActive(true); //ввімкнення другого об'єкту
    }
}
```

На кожному етапі роботи з програмою існує можливість виходу з застосунку через відповідну кнопку.

Код скрипту для виходу з програми:

```
using System.Collections;
```

```

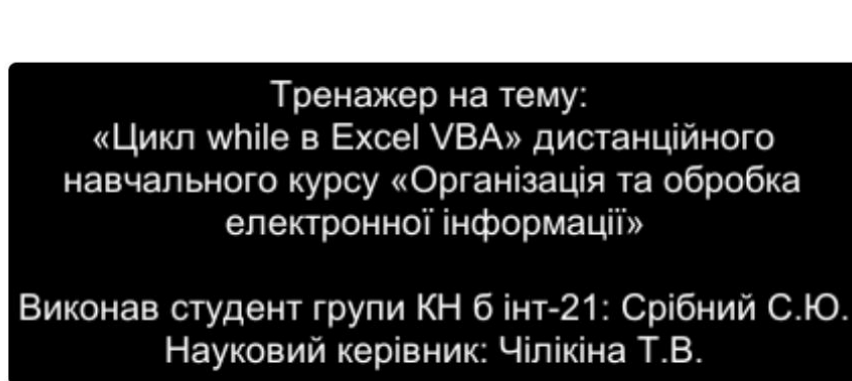
using System.Collections.Generic;
using UnityEngine;

public class QuitButton : MonoBehaviour
{
    public void QuitGame()
    {
        Application.Quit();
    }
}

```

Головне меню програми-тренажеру дозволяє перейти до роботи з теоретичними завданнями.

Цикл while в Excel VBA



Перейти до роботи

Рисунок 4.1 – Головне меню програми

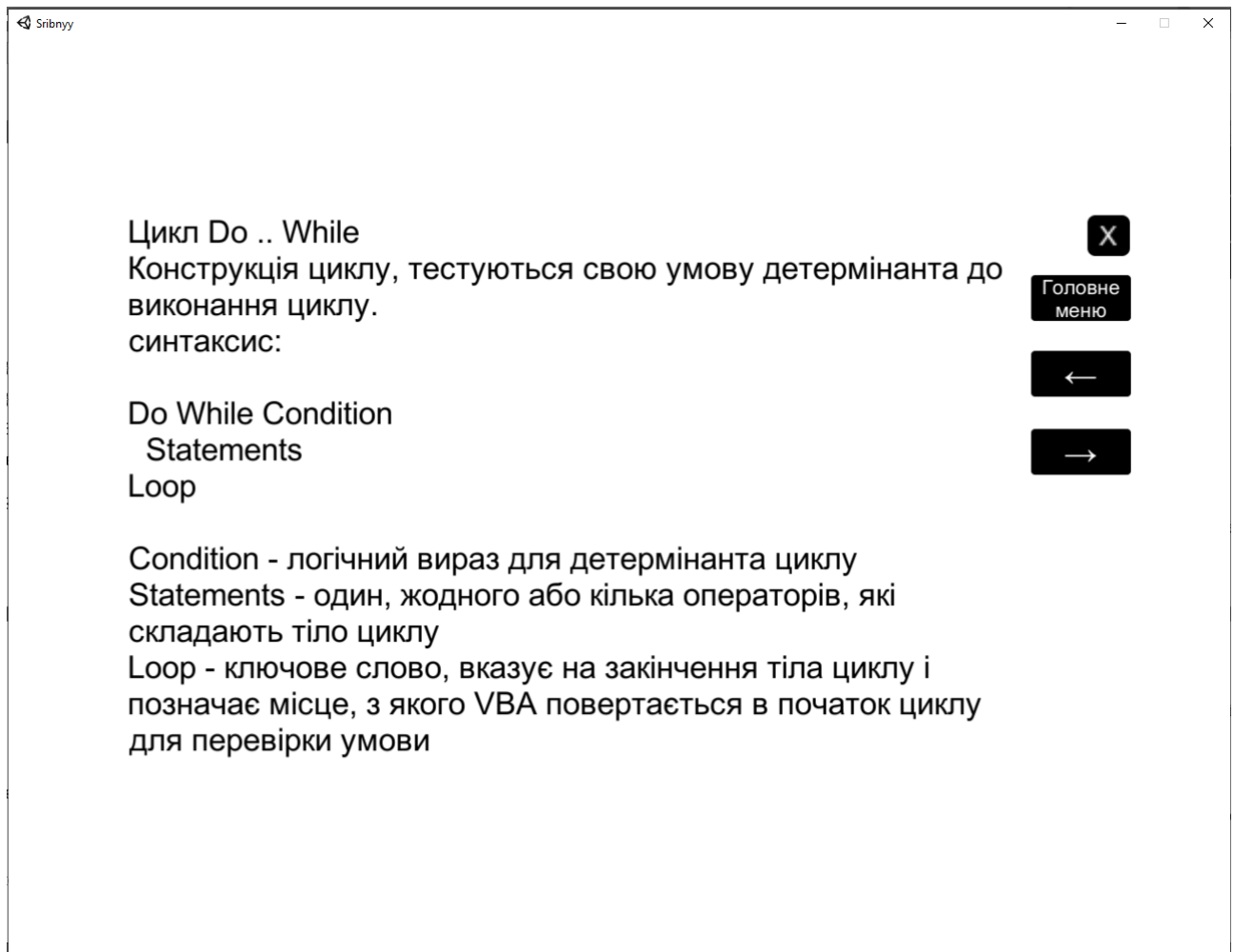


Рисунок 4.2 – Зовнішній вигляд елементів з довідковими матеріалами

Після закінчення роботи з довідковими матеріалами користувач переходить до практичних завдань тестового типу.

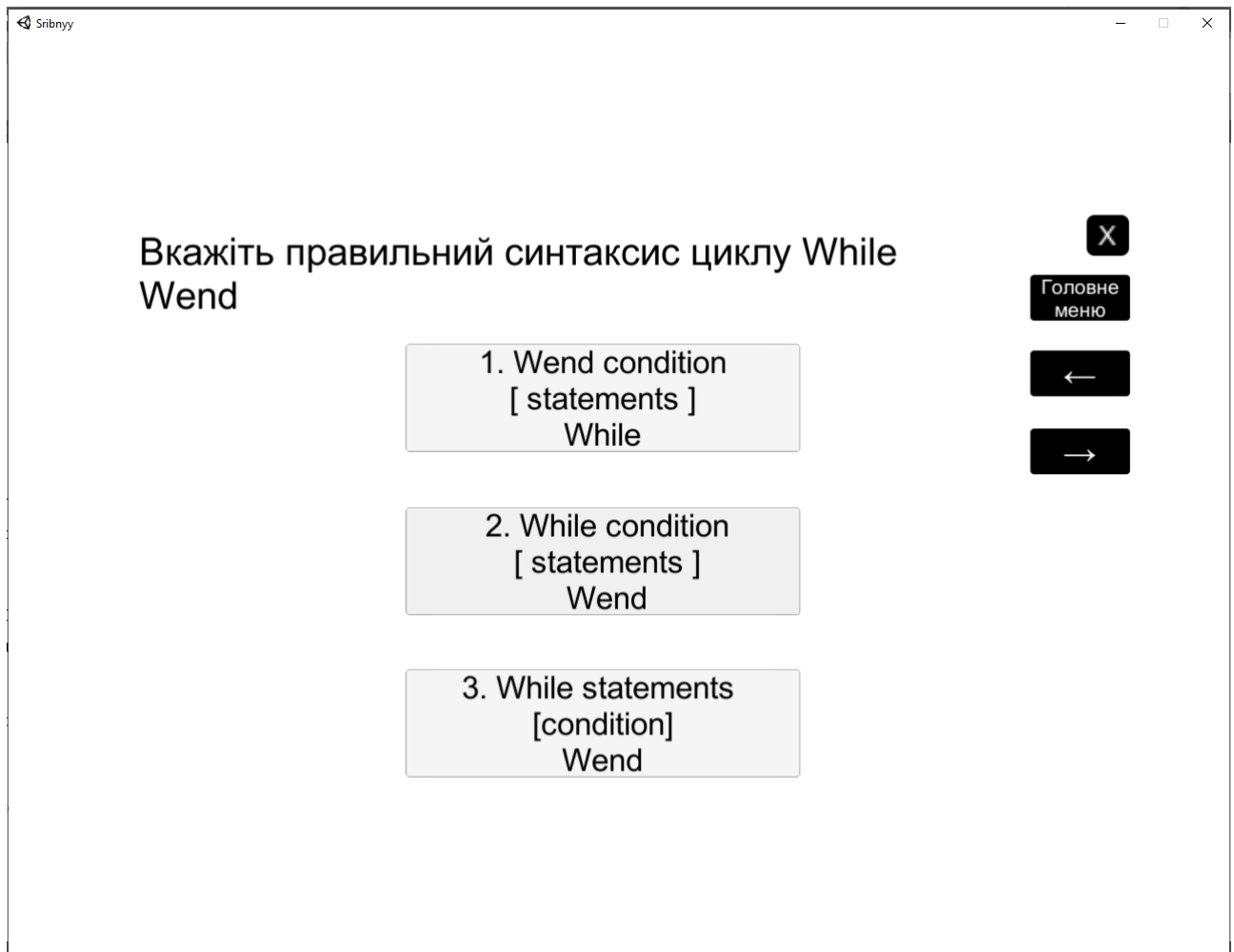


Рисунок 4.3 – Зовнішній вигляд практичного завдання тестового типу

Після закінчення роботи з практичними завданнями тестового типу користувач переходить до практичних завдань з полем для вводу.

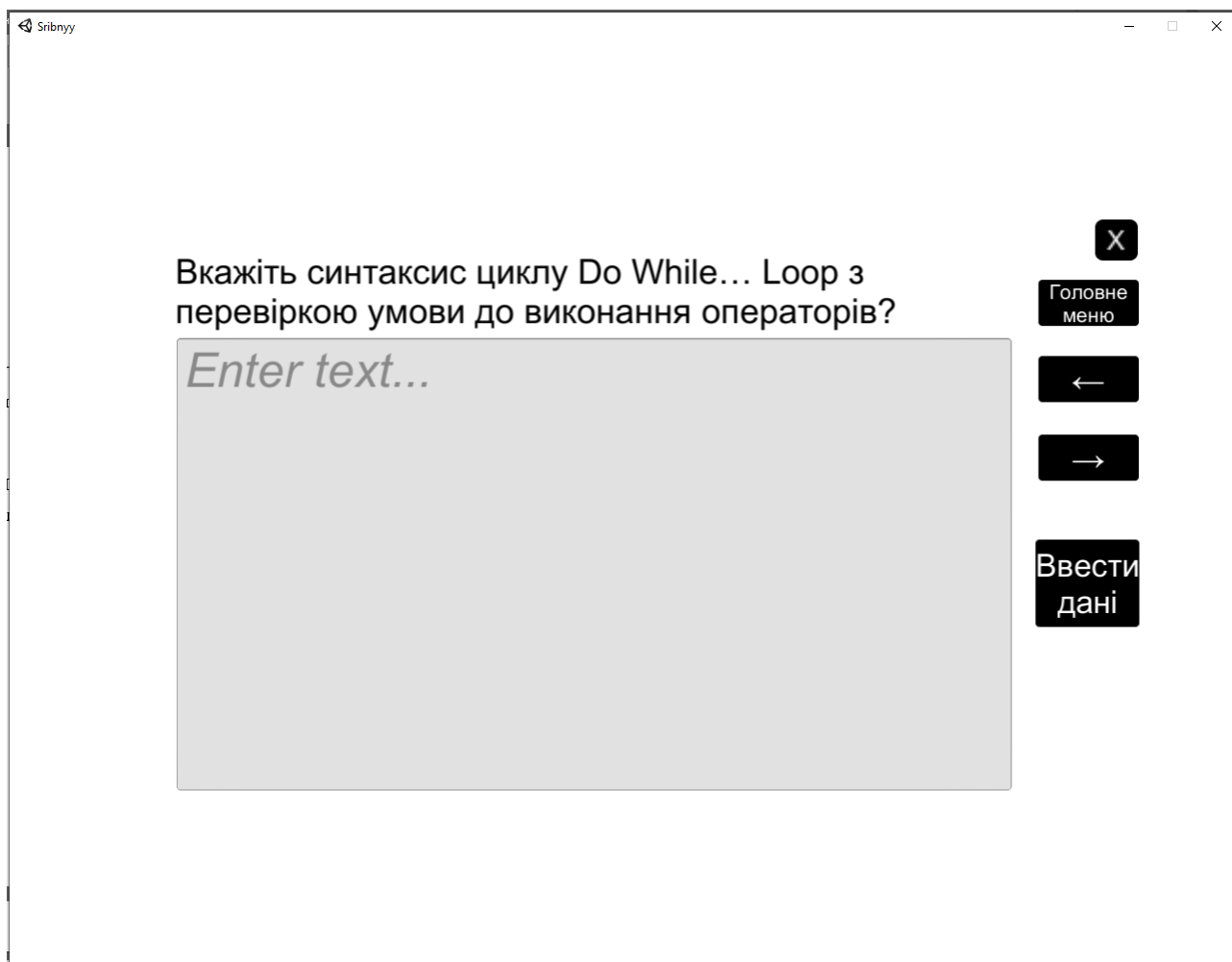


Рисунок 4.4 – Зовнішній вигляд практичних завдань з полем для вводу перед вводом даних

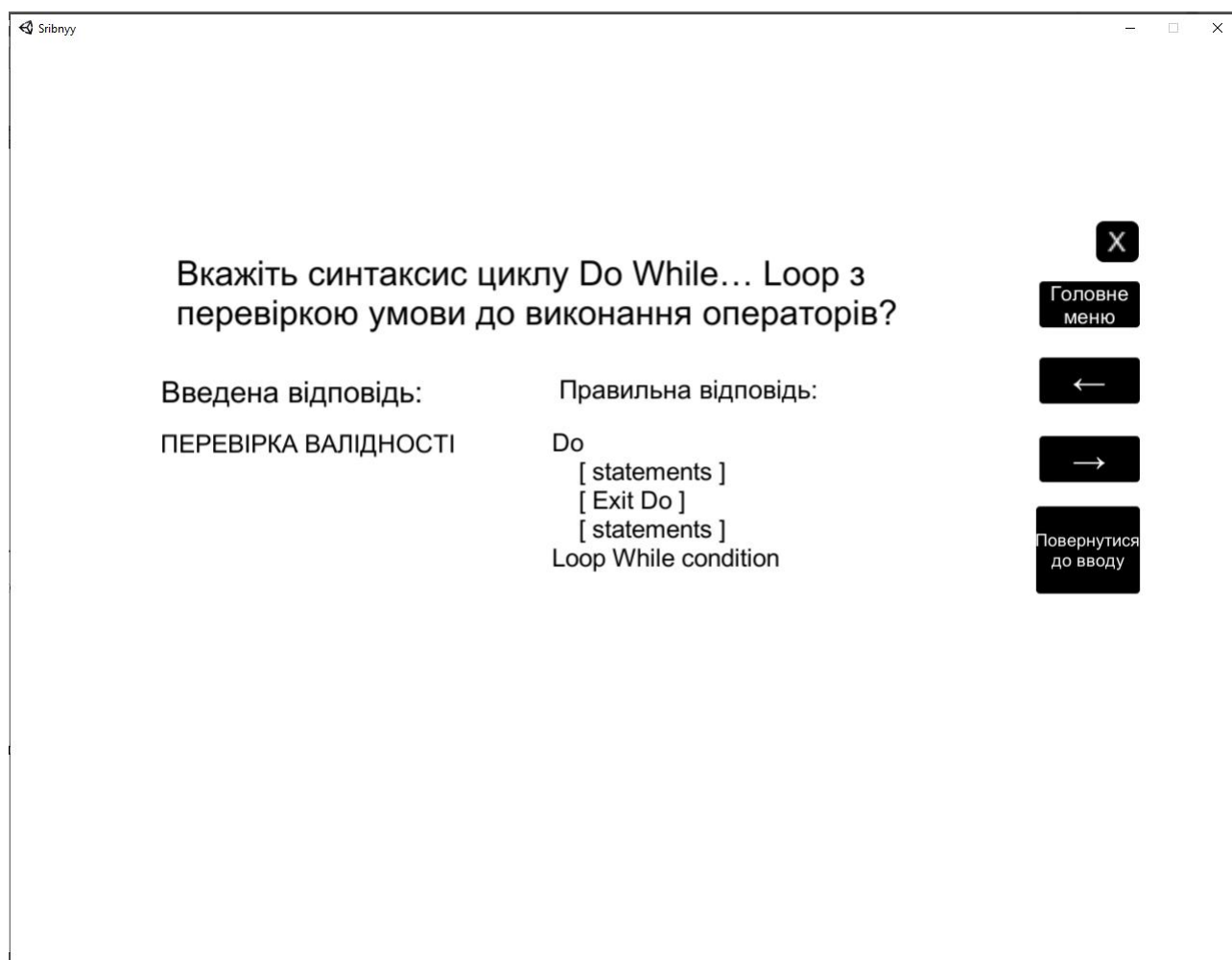


Рисунок 4.4 – Зовнішній вигляд практичних завдань з полем для вводу після вводу даних

## 4.2 Дослідження можливостей програмної реалізації

Запуск тренажера відбувається через файл .exe. Після цього є можливість обрати розмір екрану та можливість працювати в повноекранному режимі.

Головне меню програми дозволяє перейти безпосередньо до роботи з теоретичними завданнями.

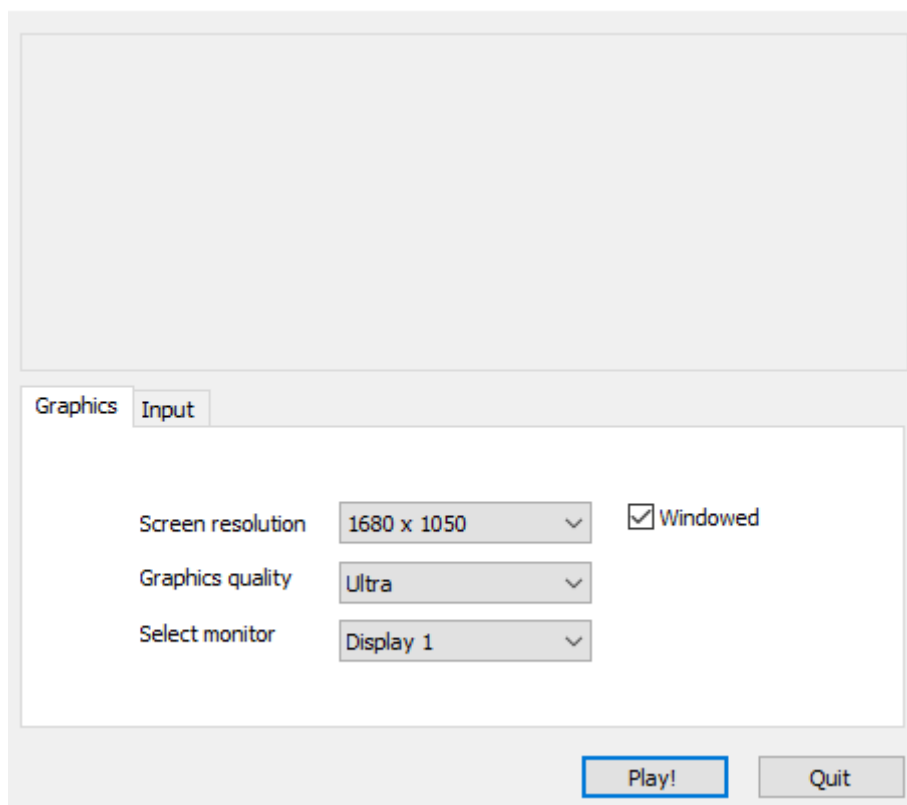
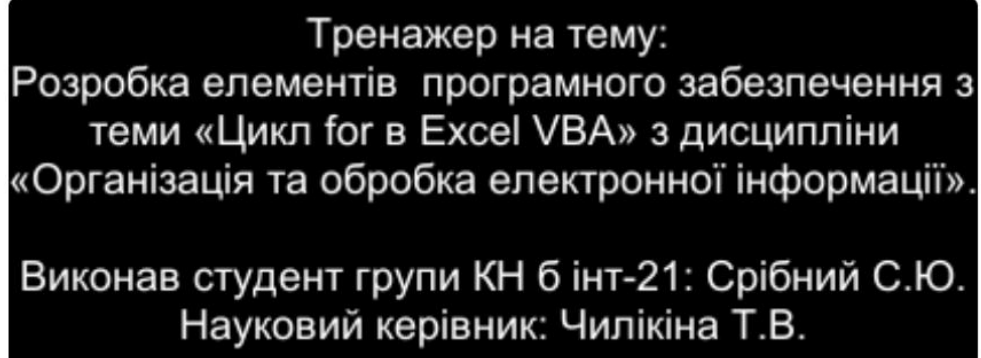


Рисунок 4.5 – Вікно для вибору налаштувань перед запуском



Перейти до роботи

Рисунок 4.6 – Запуск програми-тренажера

Кнопки для переходу між завданнями дозволяють перейти на наступний крок або повернутися до минулого.



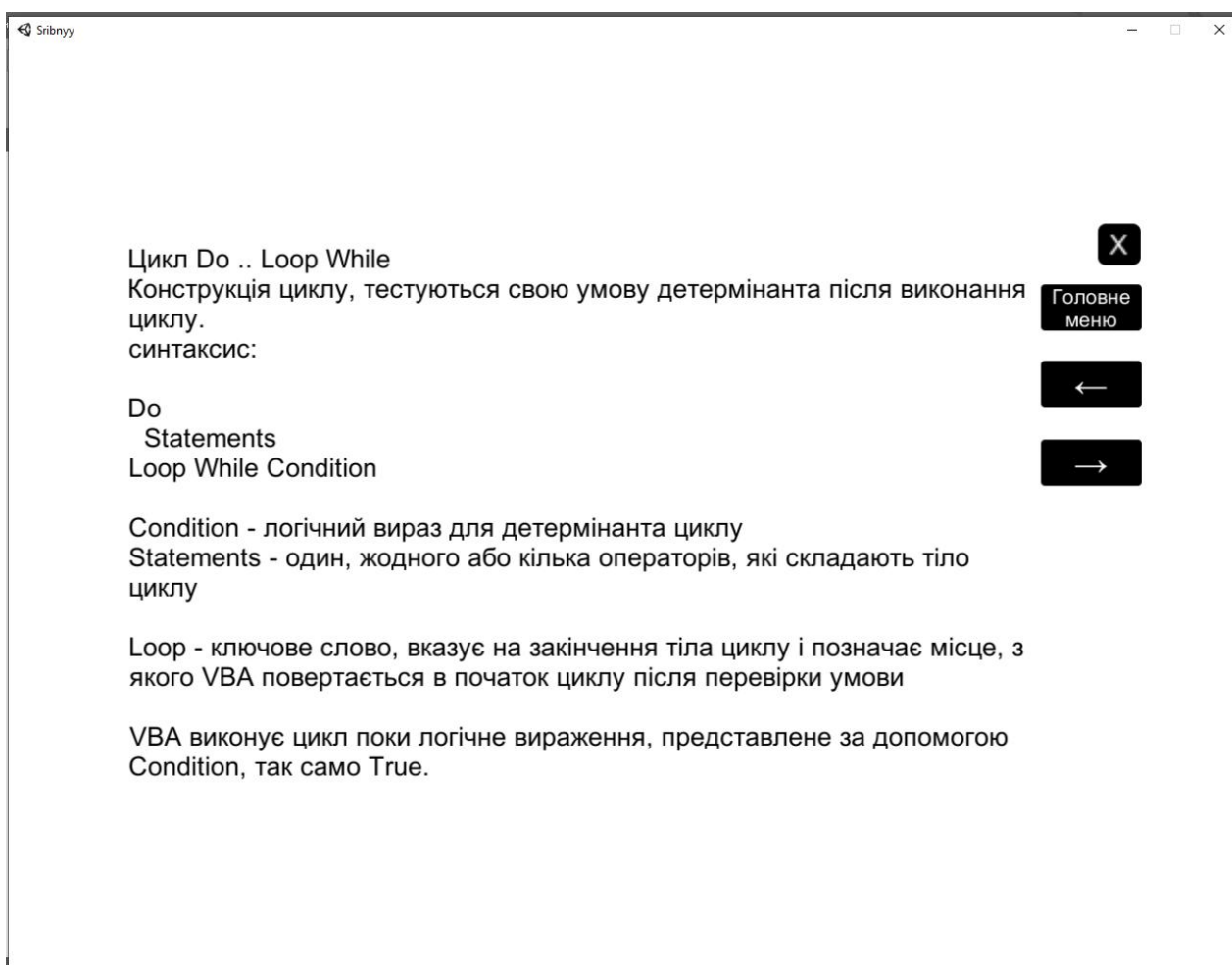


Рисунок 4.7 – Перевірка працездатності кнопок

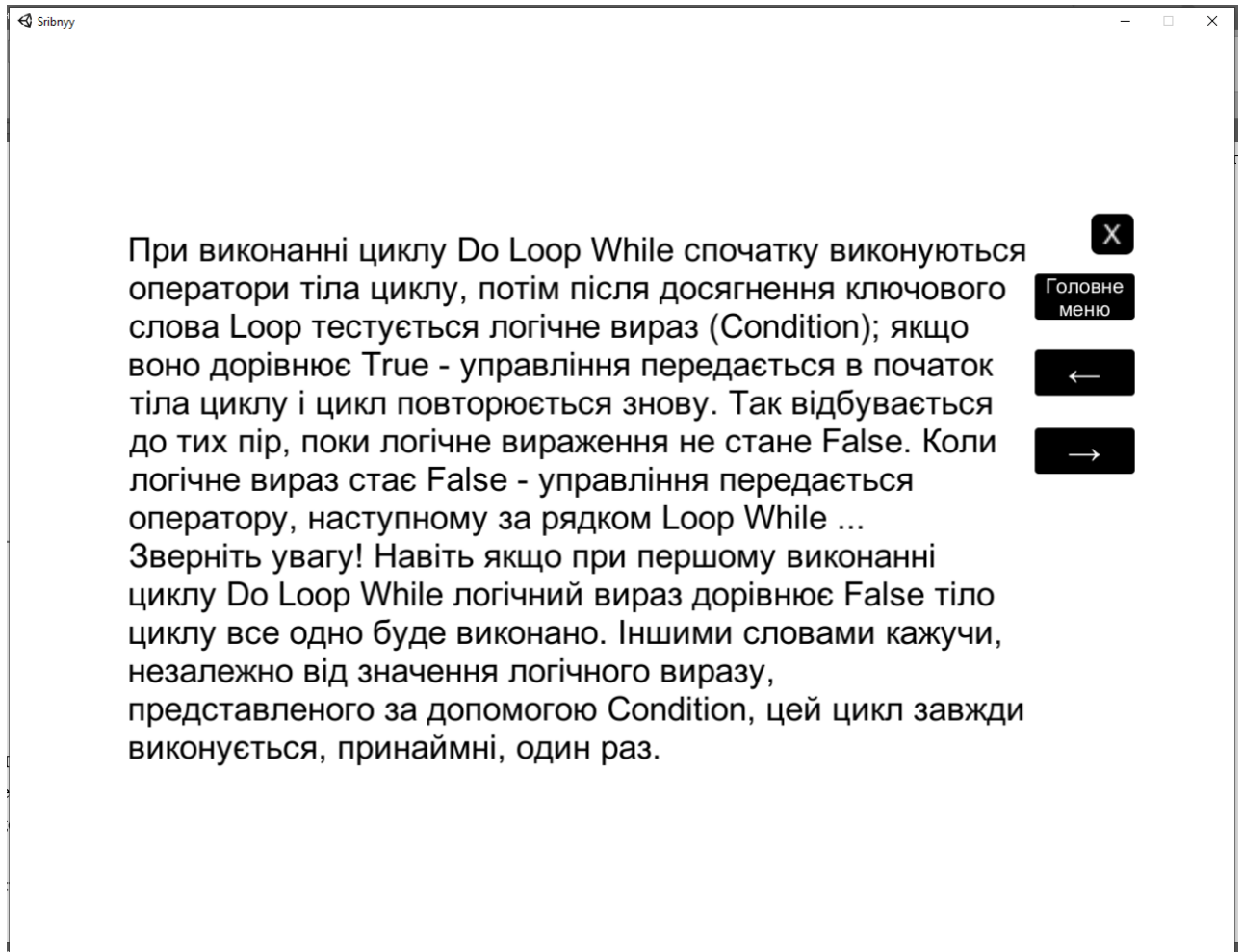


Рисунок 4.8 – Перевірка працездатності кнопок

Після завершення роботи з теоретичними матеріалами користувач отримує коротку довідку у вигляді таблиці.

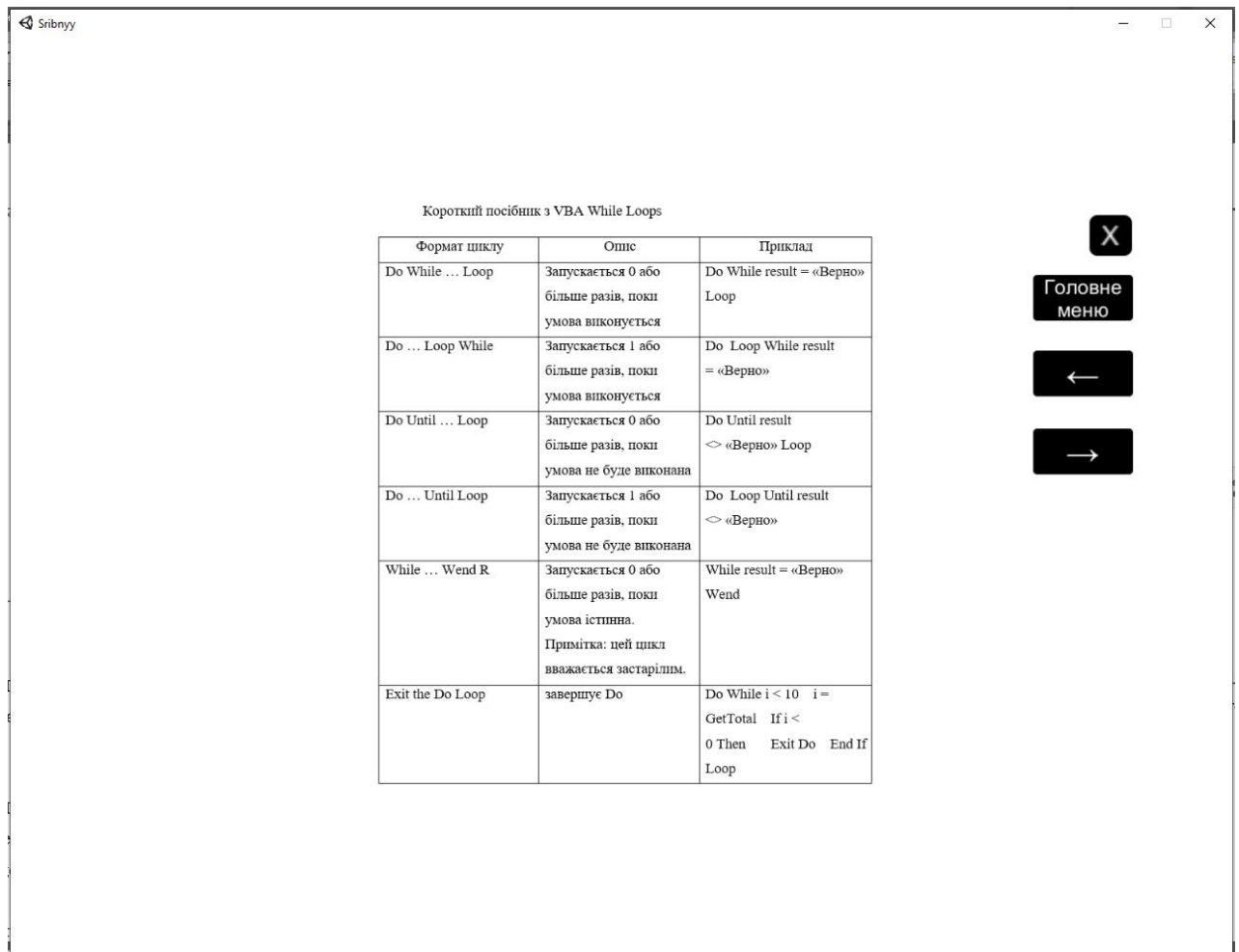


Рисунок 4.9 – Короткий посібник з VBA While Loops

При роботі з завданнями практичного типу після вводу відповіді та натиснення на кнопку вводу, відбувається перенаправлення на вікно з завданням та введеною чи правильною відповідями, також є можливість натиснути на кнопку підказки, що перенаправить користувача на типові завдання теоретичного типу.

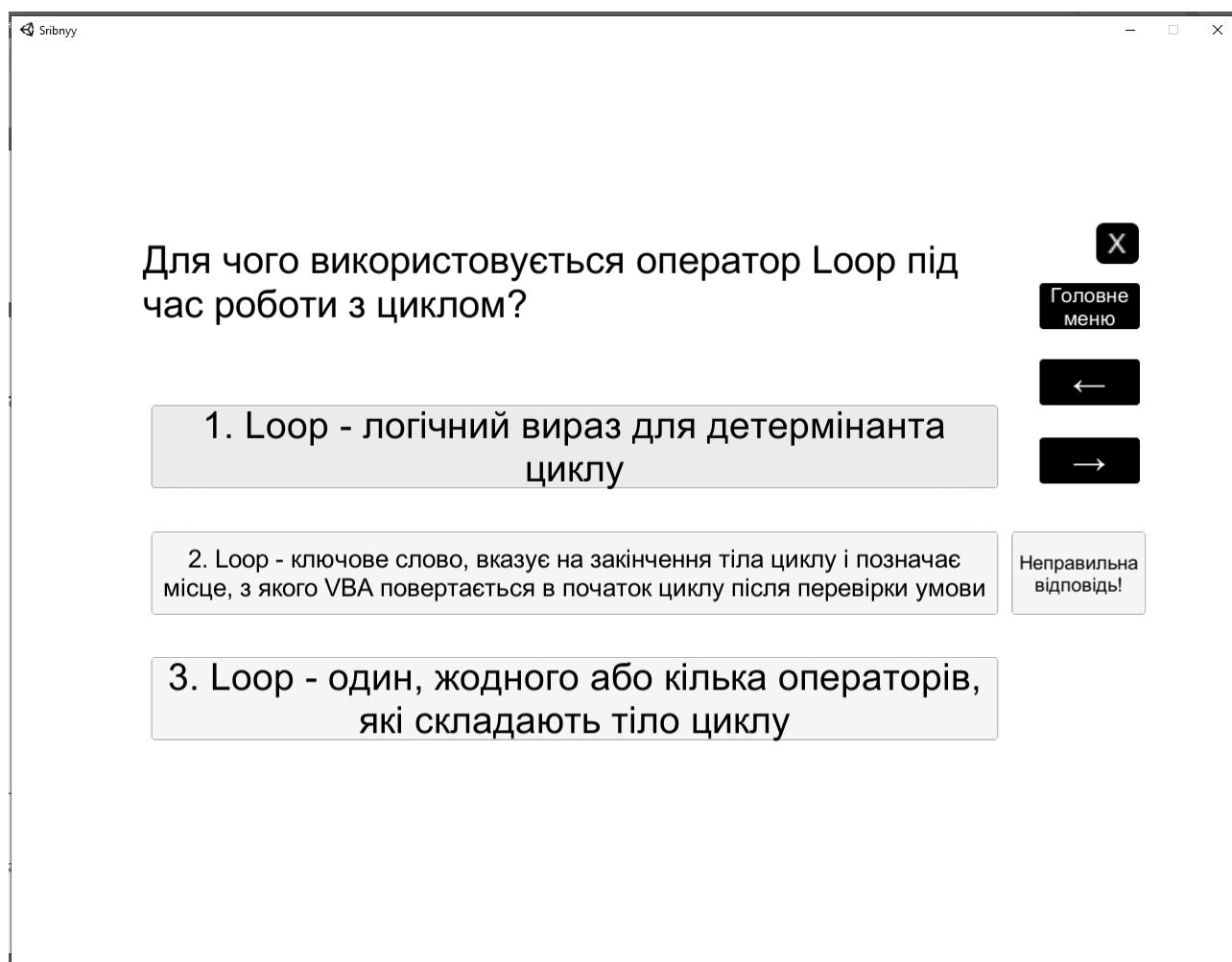


Рисунок 4.10 – Перевірка працездатності завдань практичного типу

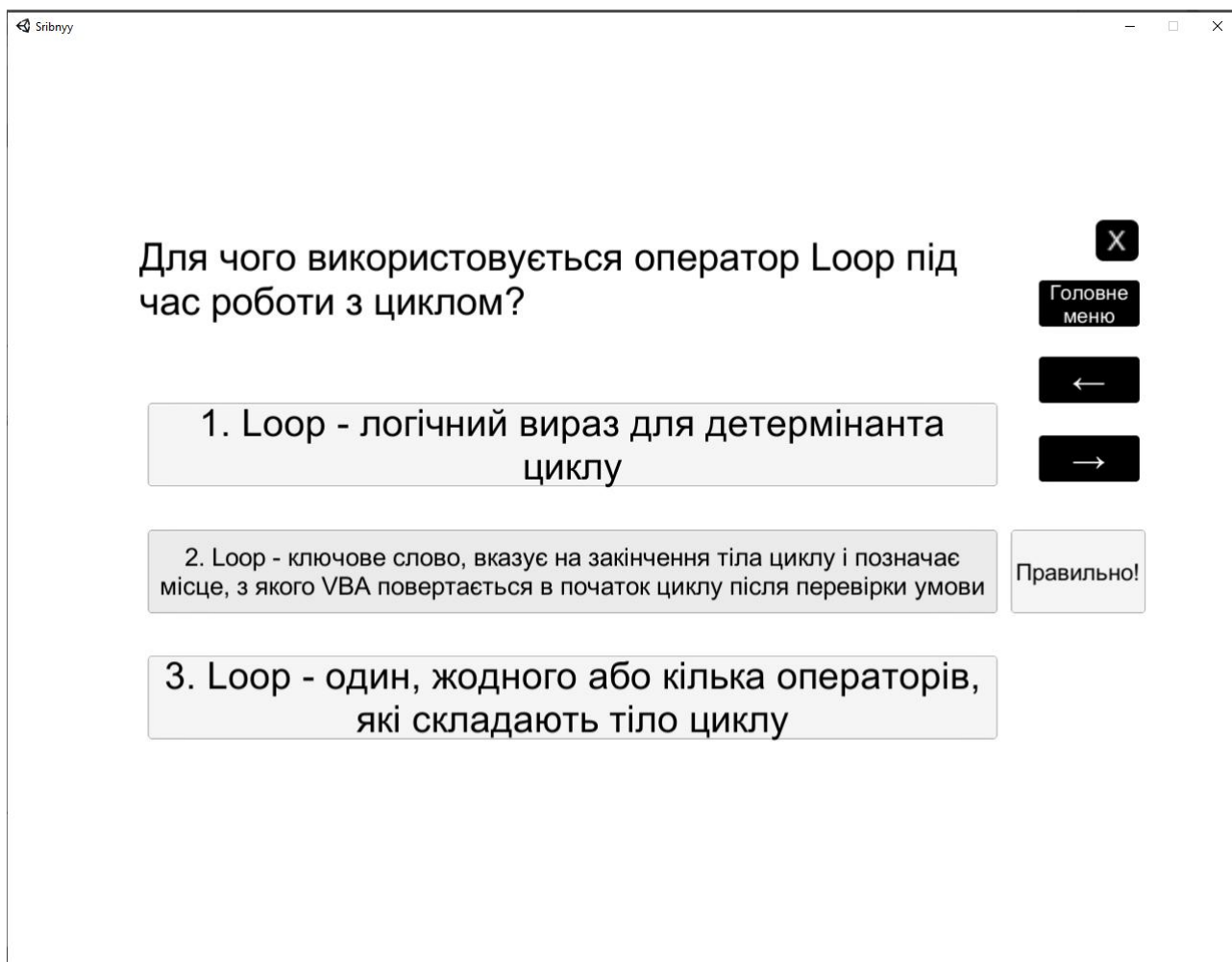


Рисунок 4.11 - Перевірка працездатності завдань практичного типу

Робота з практичними завданнями тестового типу виконується типово за зразком, після завершення роботи користувач переходить до практичних завдань з полями для вводу.

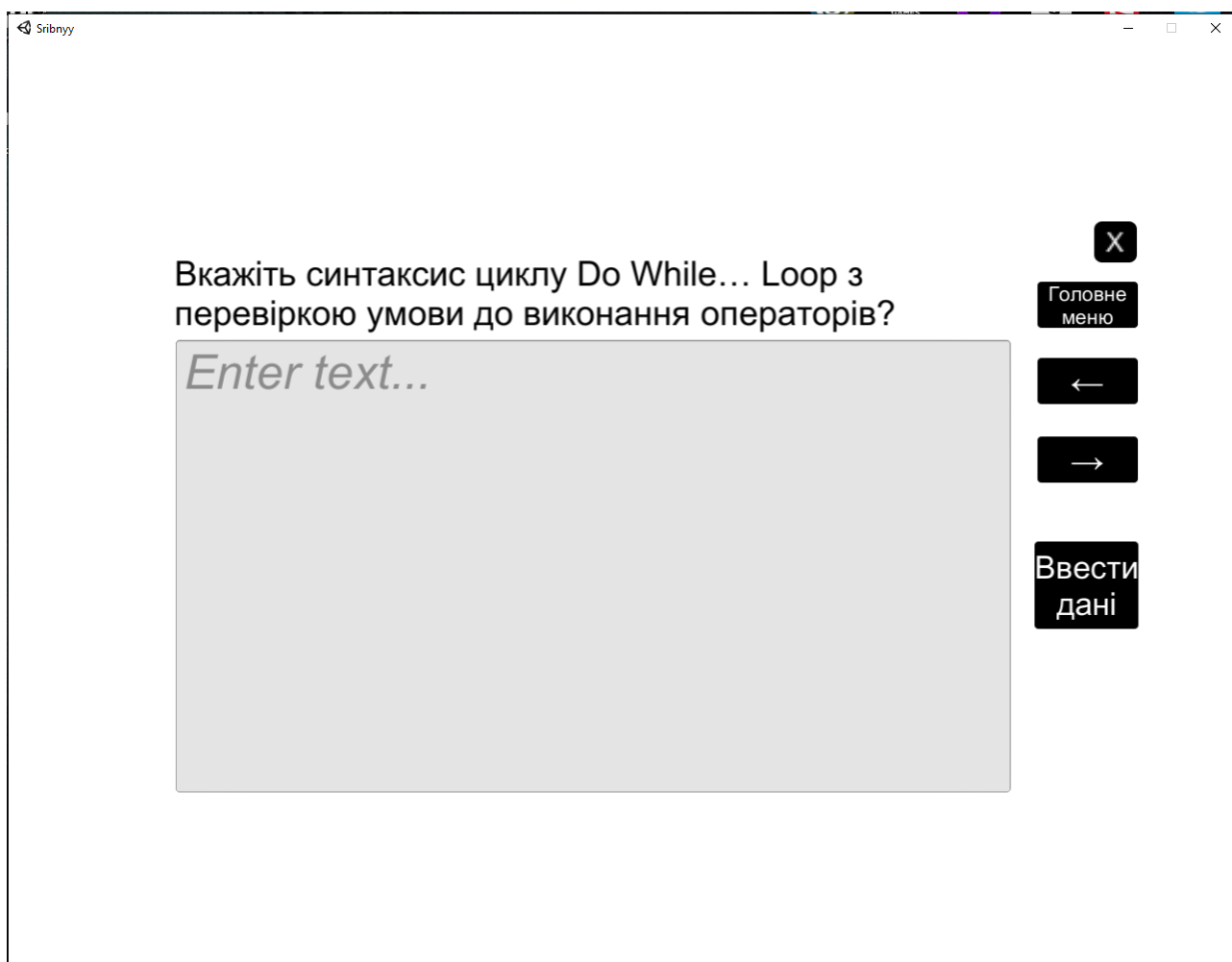


Рисунок 4.12 – Зовнішній вигляд практичного завдання з полем для вводу

Завдання з полем для вводу складається з двох частин – з вікна безпосередньо з полем для вводу і з вікна, що відкривається після вводу даних.

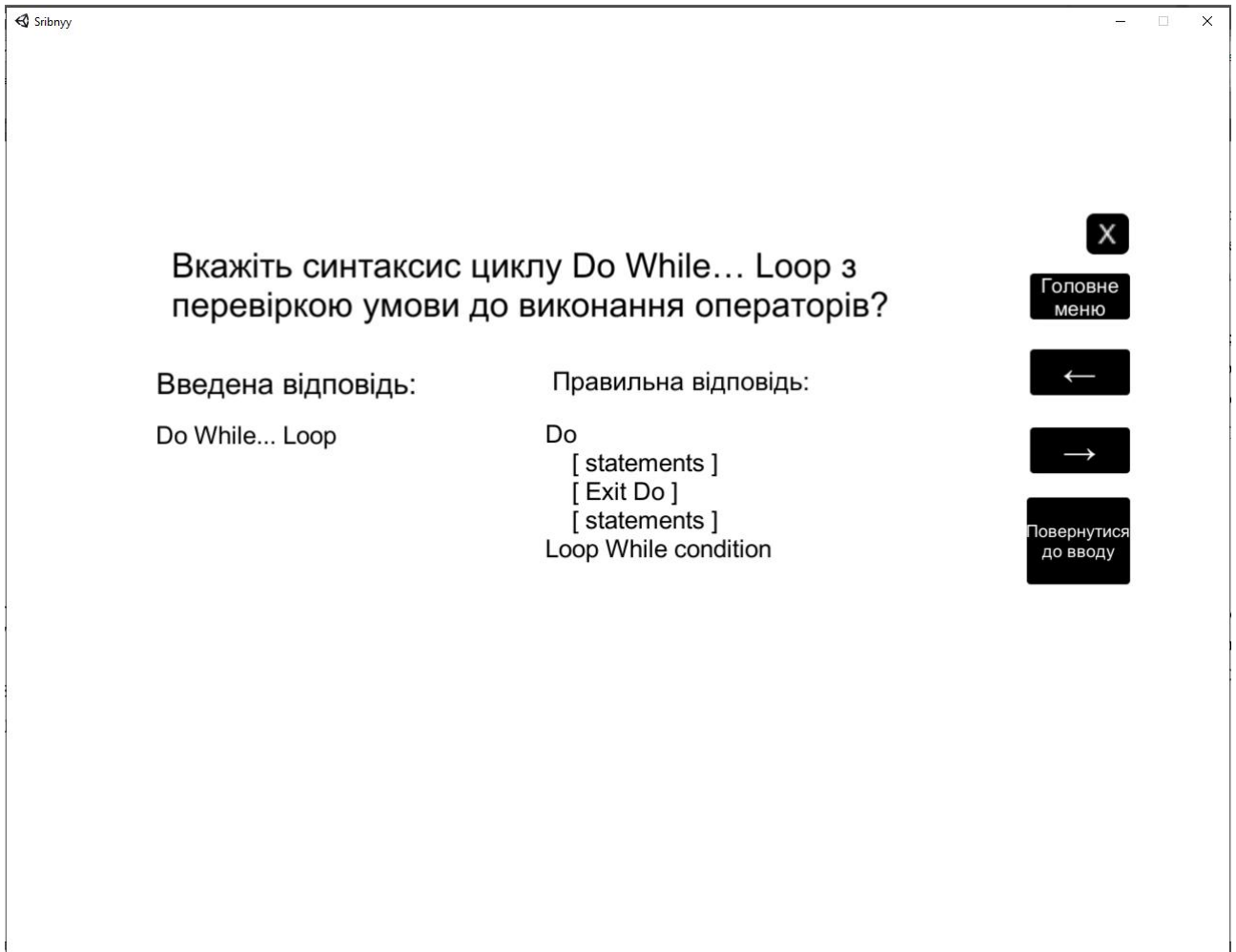


Рисунок 4.13 - Зовнішній вигляд практичного завдання з полем для вводу на другому етапі роботи з ним

Після отримання правильної відповіді у вигляді підказки користувач може повернутися до вводу та виправити свою відповідь через відповідну кнопку.

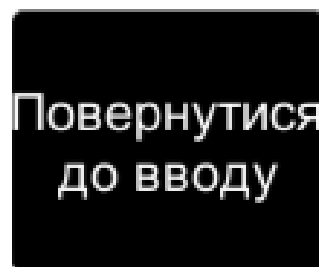


Рисунок 4.14 - Кнопка «Повернутися до вводу»

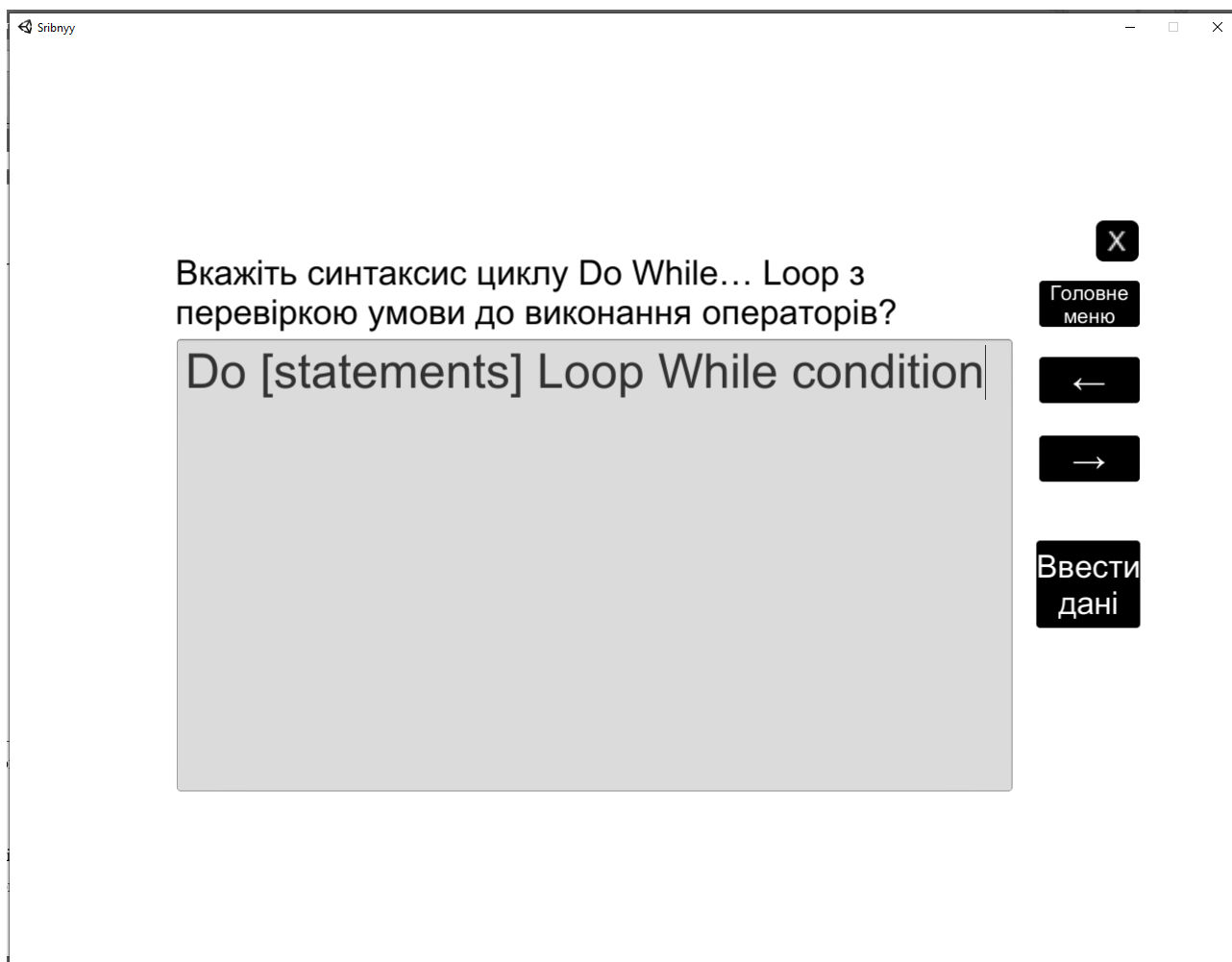


Рисунок 4.15 – Зміна відповіді

Після зміни відповіді та повторного натиснення на кнопку «Ввести дані» на другому етапі роботи з практичним завданням з полем для вводу зникає можливість виправити введену відповідь.



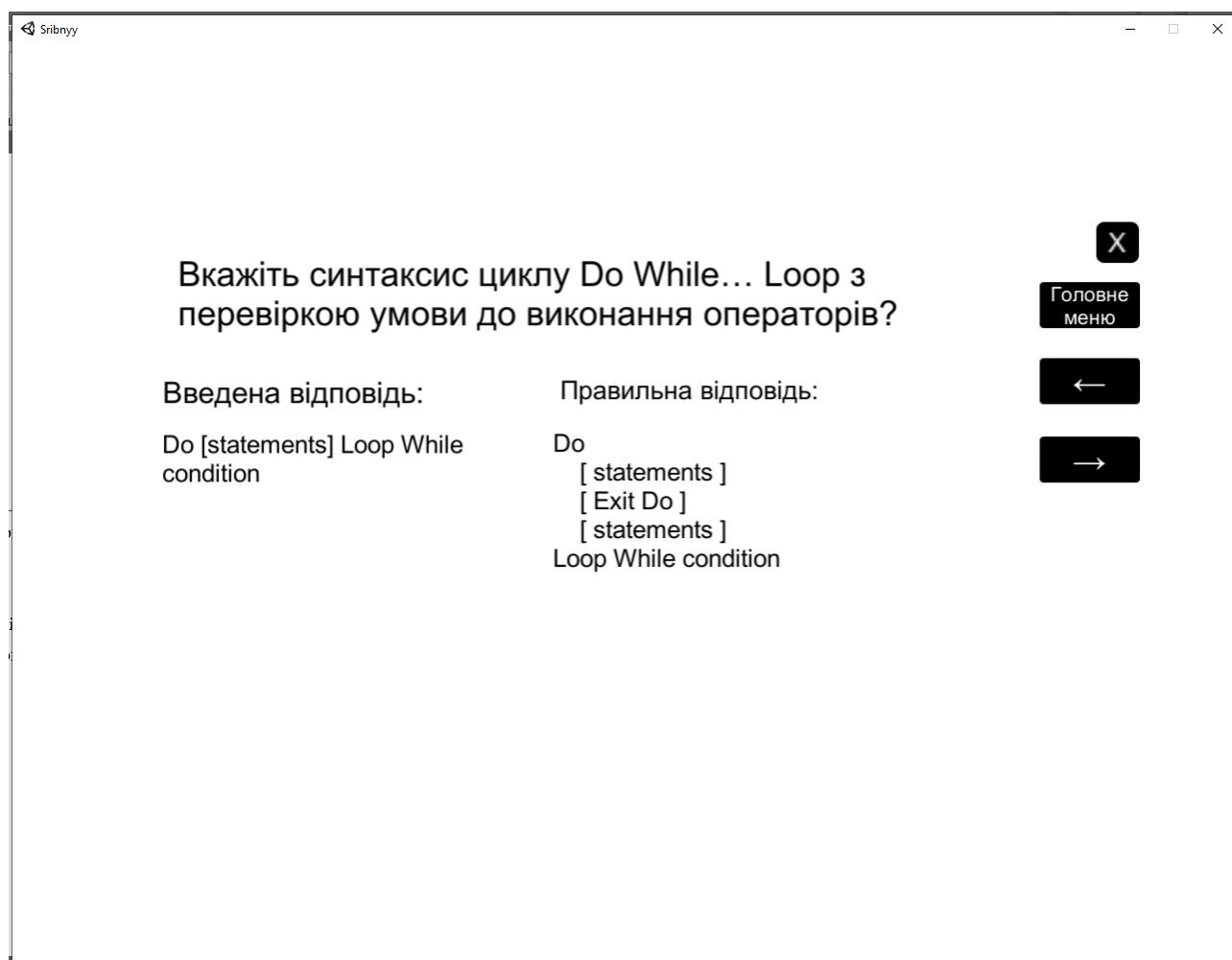


Рисунок 4.16 - Зовнішній вигляд практичного завдання з полем для вводу після зміни відповіді

## ВИСНОВКИ

Під час роботи над бакалаврською роботою було розроблено програмне забезпечення у вигляді тренажеру з теми «Цикл while в Excel VBA» для дистанційного курсу «Організація та обробка електронної інформації».

Результатами бакалаврської роботи є:

1. Вибрано методи роботи з програмним забезпеченням;
2. Виконано роботу над алгоритмом програми-тренажера;
3. Розроблено блок-схеми роботи програмного забезпечення;
4. Реалізоване програмне забезпечення у вигляді тренажеру на платформі UNITY 2018;
5. Виконано перевірку працездатності тренажеру, помилок не виявлено.

Плюсами розробленого програмного забезпечення є:

1. Зручне подання інформації;
2. Можливість виходу з застосунку з будь-якого місця роботи.

По результатам роботи створено програмне забезпечення у вигляді тренажеру з теми «Цикл while в Excel VBA» для дисципліни «Організація та обробка електронної інформації».

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кулинич М.К. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення тренажеру з теми «Злиття впорядкованих послідовностей» дистанційного навчального курсу «Алгоритми та структури даних» /Кулинич М.К. [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/9006>
2. Сивокінь О.Ю. Пояснювальна записка до дипломної роботи на тему «Логіка висловлювань» дистанційного навчального курсу «Математична логіка» та розробка його програмного забезпечення /Сивокінь О.Ю. [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/7556>
3. While... Wend – Excel VBA [Електронний ресурс].– Режим доступу до ресурсу: <https://vremya-ne-zhdet.ru/vba-excel/tsikl-while-wend/>
4. Do... While – Excel VBA [Електронний ресурс].– Режим доступу до ресурсу: <https://vremya-ne-zhdet.ru/vba-excel/tsikl-do-while-loop/>
5. OnlineTeaching –VBA [Електронний ресурс].– Режим доступу до ресурсу: <http://on-line-teaching.com/vba/lsn0117.html>
6. Excelpedia – Цикли в Excel VBA [Електронний ресурс].– Режим доступу до ресурсу: <https://excelpedia.ru/makrosi-v-excel/vse-cikly-vba>
7. Learn Unity – Unity [Електронний ресурс].– Режим доступу до ресурсу: <https://learn.unity.com/tutorials?signup=true>
8. Уроки по роботі з IDE Visual Studio [Електронний ресурс].– Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio)
9. Уроки по роботі з IDE Visual Studio [Електронний ресурс].–

Режим доступа до ресурсу: <http://www.codenet.ru/progr/vbasic/Visual-Basic-NET/>

10. IDE Visual Studio [Электронный ресурс].–

Режим доступа до ресурсу:  
[https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio)

11. Гоша Дударь, Відео-уроки по Unity, C# / Гоша Дударь [Электронный ресурс].–

Режим доступа до ресурсу:  
<https://www.youtube.com/watch?v=GGsOU7sP0r4>

## ДОДАТОК А

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;

    public InputField inputedTxt;
    public Text ShowInpTxt;

    public void Next()
    {
        Input.SetActive(false);
        AfterInput.SetActive(true);

        ShowInpTxt.text = inputedTxt.text;
    }
}

public class QuitButton : MonoBehaviour
{
    public void QuitGame()
    {

```

```

        Debug.Log ("QUIT!!");
        Application.Quit();
    }

```

```

void Start()
{
    Button btn = nxtButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}

```

```

void TaskOnClick()
{
    Theme1.SetActive(false);
    Theme2.SetActive(true);

    Debug.Log("You touched this button.");
}

```

```

void Start()
{
    Button btn = menuButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}

```

```

void TaskOnClick()
{
    Theme6.SetActive(false);
    MainMenu.SetActive(true);
}

```

```
    Debug.Log("You touched this button.");  
}
```

```
public void Next()  
{  
    Input.SetActive(false);  
    AfterInput.SetActive(true);  
  
    ShowInpTxt.text = inputdTxt.text;  
}  
}
```